

# Context-dependent reconfiguration of autonomous vehicles in mixed traffic

José Miguel Horcas<sup>1</sup>  | Julien Monteil<sup>2</sup> | Mélanie Bourroche<sup>2</sup> | Mónica Pinto<sup>1</sup> | Lidia Fuentes<sup>1</sup> | Siobhán Clarke<sup>2</sup>

<sup>1</sup>CAOSD Group, Universidad de Málaga, Andalucía Tech, Spain

<sup>2</sup>Distributed Systems Group, School of Computer Science and Statistics, Trinity College Dublin, Ireland

## Correspondence

José Miguel Horcas, Bulevar Louis Pasteur, 35. ETSII Lab. 3.3.3, Campus de Teatinos, Malaga 29071, Spain.  
Email: horcas@icc.uma.es

## Funding information

Magic, Grant/Award Number: P12-TIC1814; HADAS, Grant/Award Number: TIN2015-64841-R

## Abstract

Human drivers naturally adapt their behaviour depending on the traffic conditions, such as the current weather and road type. Autonomous vehicles need to do the same, in a way that is both safe and efficient in traffic composed of both conventional and autonomous vehicles. In this paper, we demonstrate the applicability of a reconfigurable vehicle controller agent for autonomous vehicles that adapts the parameters of a used car-following model at runtime, so as to maintain a high degree of traffic quality (efficiency and safety) under different weather conditions. We follow a dynamic software product line approach to model the variability of the car-following model parameters, context changes and traffic quality, and generate specific configurations for each particular context. Under realistic conditions, autonomous vehicles have only a very local knowledge of other vehicles' variables. We investigate a distributed model predictive controller agent for autonomous vehicles to estimate their behavioural parameters at runtime, based on their available knowledge of the system. We show that autonomous vehicles with the proposed reconfigurable controller agent lead to behaviour similar to that achieved by human drivers, depending on the context.

## KEYWORDS

autonomous vehicles, car-following model, dynamic software product line, reconfiguration, traffic quality

## 1 | INTRODUCTION

Drivers' behaviour often changes during a journey because of several environmental factors such as the typology of the roads (motorways, high speed roads, arterials, and local streets), the type of road surface (asphalt, concrete, and gravel roads), and weather conditions (dry, rain, and snow). A human driver takes into account all these factors and adapts his or her driving style accordingly.<sup>1</sup>

In mixed-traffic environments, ie, where autonomous vehicles share the roads with conventional vehicles, autonomous vehicles should also adapt to these contextual changes but without deteriorating traffic safety and efficiency, as autonomous vehicles' behaviours that are overly different to human driven vehicles might lead to hazardous and chaotic situations.<sup>2</sup> The behaviour of an autonomous vehicle will be altered at runtime by modifying/reconfiguring the parameters of the car-following model (eg, desired speed, desired time headway, acceleration, and braking) or even by changing the model itself (eg, the car-following model or the lane-changing protocol). For instance, lane-changing handlers in charge of controlling vehicles on motorways mostly rely on minimising decelerations (eg, the MOBIL model<sup>3</sup>), whereas a lane-changing handler in urban traffic relies on the allocation of vehicles' neighbours and the management of conflict areas (eg, the Vertigo model<sup>4</sup>). In any situation, the dynamic selection of the most appropriate set of model and parameters must help to increase traffic-flow safety and efficiency. However, the literature has just started to investigate the applicability of various car-following models under varying environmental contexts such as adverse weather conditions.<sup>5,6</sup> Driving scenarios (eg, highways, intersections, and roundabouts for dry roads) are usually studied separately, and as a result, emerging control systems do not consider contextual changes of scenarios, while real-world driving involves mixing different scenarios on the same route.

Moreover, the parameters values for an autonomous vehicle controller are specific to each context and need to be taken into account at runtime: If no behavioural change is introduced, the sudden appearance of adverse weather conditions will cause autonomous vehicles to have unsafe time headways, and the sudden improvement of weather conditions will cause autonomous vehicles to be overly cautious, having a detrimental effect on

traffic efficiency. Car-following models, which can be collision-free or collision-prone, time continuous or time delayed, incorporate a high degree of behavioural variability in the longitudinal dynamics. Their behavioural parameters (desired speed, desired time headway, maximum acceleration, jam distance, etc) depend on the vehicle characteristics and the driving style and are context-dependent. The impact of behavioural parameters on traffic safety and efficiency under specific weather conditions can be assessed via traffic quality indicators (mean speed, total time spent in the network, local densities, time to collision, etc).

In this paper, we demonstrate the applicability of a vehicle controller agent for autonomous vehicles that is adaptive to contextual changes, embodied by changes in weather conditions (dry conditions, light rain, and heavy rain). The controller agent automatically reacts to changes in the context by reconfiguring the behavioural parameters of the car-following model used, to maintain a high degree of traffic quality (efficiency and safety) with varying weather conditions. The controller agent has been developed following a dynamic software product line (DSPL)<sup>7</sup> approach, which deals with the high degree of variability in an autonomous vehicle's behaviour, the environmental context, and the traffic quality. Moreover, using a DSPL, our approach generates specific configurations for the vehicle controller agent for each particular context, at runtime. Software product lines (SPLs)<sup>8,9</sup> are traditionally used to exploit and handle the commonalities and variations of applications in different domains. However, variability in vehicle controllers has not received much attention.<sup>6</sup> For example, there is no approach that integrates the variability of context changes, a vehicle's behaviour, and the indicators of traffic quality, as part of the variability management of an SPL.

The paper is structured as follows. First, in Section 2, we introduce the background information and describe the relationships between weather conditions and the behavioural parameters of car-following models. We also present the challenges we address in this paper. In Section 4, we design an estimate of the ideal behavioural parameters for autonomous vehicles, assuming that autonomous vehicles have full knowledge of the vehicles' positions and speeds in the system of vehicles. Then, in Section 5, we present our reconfigurable vehicle controller. We focus on realistic conditions, where autonomous vehicles only have partial information about the system variables, and perform online optimisation for each autonomous vehicle within a model predictive controller agent. Section 6 assesses the relevance of our algorithms, and we compute indicators of traffic-flow efficiency (travel time and homogeneity of speeds) and of traffic-flow safety (eg, time to collision and post-encroachment time). We evaluate these indicators with and without reconfiguration, showing the relevance of our approach for increasing the quality of traffic flow. Section 3 discusses previous work in the literature. Finally, Section 7 concludes the paper.

## 2 | BACKGROUND INFORMATION AND MOTIVATION

In this section, we first introduce common terms relating to autonomous vehicle controllers and car-following models, the main behavioural parameters of car-following models, and their relationships with weather conditions. We also provide a background to DSPLs. Then, in section 2.4, the motivation of our approach is presented as a set of challenges we address in this paper.

### 2.1 | Car-following models and behavioural parameters

An “autonomous vehicle controller” is a system with the power and ability for self governance in the performance of control functions of a vehicle under significant uncertainties in the environment, without external intervention, over extended periods.<sup>10</sup> The behaviour of each individual vehicle (“microscopic traffic”) is defined by the traffic model (eg, a car-following model) of the autonomous vehicle controller, to see its implications on the whole traffic system (“macroscopic traffic”).

In a “car-following model,” vehicles update their cinematic according to their perception of a leading vehicle's behaviour. There are 2 main types of car-following models: “time continuous” and “time delayed.” Time continuous models relate to empirical models that model the acceleration trajectories of vehicles as a continuous function of time, whereas time delayed models relate to physical models that try to capture reaction times and stimulus-response mechanisms. In addition, car-following models can be “collision-free” or “collision-prone” in the sense that collisions between 2 vehicles (crashes) are prohibited or possible.

Our approach is independent of the car-following model used, so it is possible to use any car-following model like the Intelligent Driver Model (IDM),<sup>11</sup> the Gipps model,<sup>12</sup> or optimal velocity with relative velocity (OVRV).<sup>13</sup> Concretely in this paper, we have selected the IDM model because it is collision-free, it describes shock-wave patterns and any other traffic regime very well, and it is identifiable (the same set of behavioural parameters can describe the behaviour of one driver in different traffic regimes). For these reasons it is one of the most robust existing car-following model, and very suitable for our analysis. Moreover, depending on the traffic conditions, the autonomous vehicles may use a different car-following model. For instance, the Gipps model can be more appropriate in the event of car accidents, since the Gipps model is specially designed to prioritise safety over efficiency,<sup>12</sup> while the OVRV model is more appropriate to suppress traffic jams and increase traffic capacity.<sup>13</sup>

Car-following models have a lot of configurable parameters that determine the behaviour of the autonomous vehicles such as speed, acceleration, deceleration, time headway, and jam distance. As our objective is to provide a vehicle controller that is adaptive to contextual changes, we focus on the relationship between car-following models parameters and different weather conditions, in particular on rainy adverse conditions. The main parameters that affect the behaviour of autonomous vehicles to environmental changes are *speed* ( $v$ ), the desired safe *time headway* ( $Th$ ), and the maximum *tolerated acceleration* ( $acc$ )—see Section 3. The time headway is the time that elapses between the arrival of the leading vehicle and the following vehicle at a designated point. The tolerated acceleration is the maximum acceleration that is tolerated by the driver/vehicle in a particular environmental context.

## 2.2 | Relationships between car-following behavioural parameters and weather conditions

It was recently shown in Monteil et al<sup>14</sup> and Punzo et al<sup>15</sup> that, under dense traffic conditions with speeds (eg, around 17 m/s in highways) that are far enough from the desired speeds (ie, the maximum speed limit, here, 33 m/s), only 2 parameters are sensitive and therefore need to be estimated: the desired time headway  $Th$  and the maximum tolerated acceleration  $acc$ . These 2 parameters are therefore the parameters we will consider for reconfiguration at runtime in this paper.

We make use of the NGSIM dataset (Next Generation Simulation of Traffic Analysis Tools Program)\* to build real assumptions about how the parameters  $Th$  and  $acc$  can vary under different weather conditions. We get fitted values for  $Th$  and  $acc$  from the NGSIM dataset, by filtering the data, and by fitting each observed trajectory data with car following–based simulated trajectory data reacting to the leading vehicles' trajectory data. Then, once we have estimated values for the 2 parameters, we consider that those parameters are impacted by weather conditions in the following way, in accordance with the literature (see Section 3). The desired time headway is computed for light rain conditions with an expected variation of 50% and with an expected variation of 180% for heavy rain conditions; the maximum tolerated acceleration is computed for light rain conditions with a expected variation of 10% and with an expected variation of 45% for heavy rain conditions. We are aware that these are strong hypotheses to be introduced, but they enable the approximation of realistic behaviours, and therefore, the controllers to be reconfigured are considered to be close enough to the observed real traffic situations.

## 2.3 | DSPLs and adaptive traffic behaviour

An SPL is “a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way.”<sup>9</sup> The SPLs' engineering process is in charge of generating several configurations of the same system at design time. A DSPL brings this engineering process to runtime, where a single system is able to adapt its behaviour at runtime.<sup>7,16</sup>

“Variability modelling” is the main activity of both SPLs and DSPLs, where the commonalities and variabilities of the system are specified. Then, the SPL engineering process generates products by selecting specific characteristics specified in the variability model. In an SPL, the software engineer binds the variation points to the software architecture at design time, considering the requirements of the intended system. The “software architecture” defines the components that model the behaviour of the application and that communicate with each other using connectors.<sup>17</sup> The variable parts of the software architecture are explicitly specified in the variability model at an abstract level. In a DSPL, the variability model describes the potential range of variations that can be produced at runtime for a single system—ie, the dynamic variation points. Therefore, the software architecture supports all possible adaptations defined by the set of dynamic variation points.<sup>7</sup>

In the context of the adaptive traffic behaviour, as part of the DSPL, the engineer must define (1) the variability of the car-following models (eg, all configurable parameters) and the variability of the traffic context and the traffic quality indicators; (2) the range of potential adaptations supported by the system in terms of architectural components, ie, the software architecture of the autonomous vehicles; (3) the context changes that may trigger an adaptation, ie, the criteria to initiate a reconfiguration or decision making process (eg, a change in the weather conditions); and (4) the set of possible reactions to context changes that should be supported the system (eg, change the acceleration and time headway parameters).

## 2.4 | Challenges

This subsection describes the main challenges to be addressed in the development of our reconfigurable controller agent. Here, we introduce an approach to address each challenge as well as the reason for using a DSPL.

**Challenge 1. To model and manage the variability of the autonomous vehicles' behaviour and of the traffic context.** As previously discussed, car-following models present a high degree of variability. They are based on different factors such as the vehicle's characteristics and driver's behaviours; can be collision-free or collision-prone, with or without explicit time-delays; and depend on parameters whose values depend on the context: desired speed, desired time headway, maximum acceleration, jam distance, among others. Moreover, the traffic context also presents a high degree of variability that can change at runtime. Real-world driving mixes different kinds of roads along the same route (eg, motorways, local streets, and roundabouts), traffic density (eg, jams and traffic flow), and adverse conditions (eg, rain and snow). Modelling this variability also implies having to model the existing relationships between the different parameters. **Our approach:** *we identify the parameters of a vehicle's behaviour and of the car-following models that are suitable for reconfiguration and explicitly model their commonalities and variabilities. We also model the variability presented in the traffic context (eg, weather conditions, road type, and road topology). To model the variability, we use the common variability language (CVL),<sup>18</sup> which is a domain-independent language for specifying and resolving variability over different models, and allows defining the links between the variability specification (features) and the assets (components, parameters, etc) of the product line—ie, the software architecture of the autonomous vehicle (section 5.1).*

\* <http://ops.fhwa.dot.gov/trafficanalysisistools/ngsim.htm>

**Challenge 2. To automatically change the behaviour of the vehicle when the current traffic context changes.** The core behaviour of the vehicles depends on the context. For instance, on motorways, a car-following model or a lane-changing handler is usually in charge of controlling the vehicle (eg, IDM model and the minimizing overall braking induced by lane change (MOBIL) model), but for urban traffic, the controller handler may rely on allocation of the neighbours and managing conflict areas (eg, the Vertigo model). Additionally, the behaviour of the vehicles is also affected by adverse conditions like rain. These kinds of context changes require the vehicle behaviour to be adapted—eg, adapting the car-following behavioural parameters such as acceleration and time headway. **Our approach:** we generate, at runtime, the new configuration of the vehicle software architecture to adapt its behaviour to the current context. To achieve this, we use a DSPL approach that allows us to change the configuration of the parameters that affect the vehicle behaviour at runtime, when the context changes. Indeed, the explicit modelling of the variability of the vehicle behaviour and of the context facilitates reasoning about them and changing them when needed. In this paper, we focus on optimising the behavioural parameters (acceleration and time headway) to provide an optimised behaviour for a particular context. To do so, we define a model predictive controller agent that performs dynamic optimisation for the parameters of each autonomous vehicle, taking into account the knowledge available at runtime (see Section 5.2).

**Challenge 3. To improve the safety and efficiency of the traffic network when autonomous vehicles are reconfigured.** Despite the high degree of variability exhibited in traffic behaviour, the autonomous vehicle controllers need to ensure safety and maintain the traffic-flow quality. **Our approach:** we use several quality traffic indicators (eg, time to collisions, travel time, and post-encroachment time) to assess safety and efficiency in the traffic network. We include this information in the DSPL to select the most appropriate indicators to evaluate the traffic quality based on the current context. The goal is to demonstrate that autonomous vehicles with our reconfigurable control agent improve the safety and efficiency in mixed-traffic networks.

## 3 | LITERATURE REVIEW AND RELATED WORK

We present a brief literature review about the effects of weather conditions on vehicles' behaviour. We also compare our approach with existing related work that focus on reconfiguration at runtime using DSPL.

### 3.1 | Effects of weather conditions on parameters

Several studies have investigated the influence of weather conditions on driving behaviour: from rain,<sup>19-23</sup> to snow,<sup>21,24,25</sup> fog,<sup>20,24</sup> or even temperature and wind.<sup>24</sup> As in this paper we have focused on rainy adverse conditions, we investigated the influence of weather conditions on the behavioural parameter considered: time headway ( $Th$ ) and acceleration ( $acc$ ). These are the parameters more affected by adverse conditions.<sup>14,15</sup> In summary, as the intensity of rain develops, acceleration is reduced and time headway is increased.<sup>26</sup>

#### 3.1.1 | Effects of rain on time headway.

Hogema<sup>22</sup> collected the percentage of vehicles with a time headway of less than 1, 3, and 5 seconds and showed the effect of rain: The percentage of vehicles with a mean time headway less than 1 second was smaller than when conditions were dry. Vehicles with a time headway of 3 seconds showed a similar, but smaller effect, while no effect of rain was found for vehicles with a time headway of 5 seconds. Billot et al<sup>19</sup> also assessed the rain impact in time headways: A drop of 12.1% of the time headway less than 2 seconds is observed under light rainy conditions, while in heavy rainy conditions, a sharper decrease of more than 18% is observed (a rise of the time headway between 2 and 10 s). Hoogendoorn et al<sup>5</sup> observed an increase of the time headway from an average of around 1.5 seconds in normal conditions to a very high value of around 3.7 seconds in the adverse weather condition. This is an indication of strong adaptation effects in longitudinal driving behaviour, and the author estimated the values for the IDM model.<sup>11</sup> Jung et al<sup>27</sup> selected time headway as one of the key weather-sensitive traffic parameters along the free-flow speed and deceleration rates. However, their research only takes into account the adjustment of the time headway value, changing it from 0.9 seconds (default) to 1.1 seconds under adverse conditions. Soria et al<sup>6</sup> also predict an increase of the time headway from 0.72 to 1.1 seconds under rainy conditions.

#### 3.1.2 | Effects of rain on tolerated acceleration.

Few studies focus on the particular effects of adverse conditions on the acceleration behaviour of a vehicle, as the acceleration behaviour of vehicles can not be observed at a macroscopic level, like desired time headways or speeds. Soria et al<sup>6</sup> estimated that, for the Gipps model,<sup>12</sup> the different rain conditions resulted in the same maximum acceleration: 3.3 m/s<sup>2</sup>. However, Hoogendoorn et al<sup>5</sup> observed that the maximum acceleration decreases substantially with rain conditions: from 1.6 to 0.5 m/s<sup>2</sup>, for the IDM model. Zhao et al<sup>28</sup> also observed that the maximum observed acceleration decreases from 4.36 to 2.50 m/s<sup>2</sup> under inclement weather. Finally, Asamer et al<sup>25</sup> showed that the desired acceleration can be reduced from 10% to 100% by adverse conditions, and in their experiments, acceleration showed a reduction to 71% of original acceleration settings.

### 3.2 | Reconfiguration of vehicle behaviour

Two similar reconfiguration approach to ours are presented in Asamer et al<sup>25</sup> and in Cetina et al.<sup>29</sup> Asamer et al<sup>25</sup> propose preselecting several parameters and choosing value ranges for those parameters to deal with the complexity of a large search space when all possible parameters of the

car-following model are changed during calibration, and not all the combinations can be evaluated in an acceptable time. Cetina et al<sup>29</sup> propose an approach for the design of pervasive SPLs that are reconfigured according to changes in the environment, although they do not focus on changing vehicle behaviour. They use feature models to specify the variability and event condition action (ECA) rules to trigger the reconfiguration process. To cope with the complexity of the variability, the SPL is pruned at design time, removing those model elements that are related to scenarios that are not useful and therefore limit the configuration space. In our approach, in contrast to pruning the SPL, we delimit the possible configurations to be selected by defining Object Constraint Language (OCL)<sup>†</sup><sup>30</sup> constraints over the variability model. These OCL constraints set the most appropriate parameter/model for each context, as the Asamer approach<sup>25</sup> does for the car-following parameters. This makes the generation of the configuration in our approach straightforward due to the tree constraints and cross-tree constraints of the variability model.

Although the reconfiguration approaches that focus on changing a vehicle's behaviour at runtime based on the context are scarce, there are some approaches for generic reconfiguration at runtime that use DSPL as we do. For instance, Pascual et al<sup>31</sup> present a novel approach that provides support for dynamic reconfiguration, optimising their architectural application configuration according to the available resources. Even though they focus on mobile applications, they also model the variability of the application's software architecture using SPLs, as we do. The main difference between the 2 approaches is that they take advantage of genetic algorithms to optimally resolve the application's variability, while in our approach, we use a model predictive controller agent to obtain the best parameter when the context change.

MUSIC<sup>32</sup> is an OSGi-based middleware for developing context-aware adaptive applications. It is a component-based and service-oriented approach, which principally consists of 2 different parts: the context and the adaptation middlewares. The adaptation middleware is responsible for adapting the applications, deploying the configuration that best fits the current context by evaluating a use function specified by the software architect. A difference between the MUSIC approach and our approach using DSPL is that they require the availability of all the valid configurations of the application at runtime, while our approach needs only the relationships (constraints) between the contexts and the valid configurations at runtime, while the configuration to deploy is generated on demand according the current context.

Finally, Gamez et al propose a reconfiguration mechanism that switches among different configurations at runtime.<sup>33</sup> The configurations are based on the specialisation of feature models, and the reconfiguration plans are automatically generated from the differences among them. They propagate changes in configurations at architectural level, but their approach does not support the reconfiguration of the component's parameters, as our approach does.

## 4 | IDEAL PARAMETER ESTIMATION

The main objective of this paper is to provide an automated vehicle controller adaptive to contextual changes (eg, weather conditions) that adapts a vehicle's behaviour at runtime, without deteriorating traffic-flow safety and efficiency. Before presenting our reconfigurable vehicle controller, in this section, we estimate the optimised behavioural parameters that autonomous vehicles should use in different rainy conditions if they have full knowledge of the global traffic network.<sup>34</sup>

To do so, first, we need to assess the efficiency and safety theoretically achievable by autonomous vehicle control in mixed traffic under different environmental context. We perform offline global minimisation of a defined cost function, assuming full knowledge of the vehicles' speeds and headways in the network. The goal is to obtain the behavioural parameters (safe time headway and tolerated acceleration) of autonomous vehicles associated to each environmental context (in this paper, we focus on weather conditions: normal, light rain, and heavy rain) that maximises traffic quality in terms of efficiency and safety.

$$F = \sum_{n=1}^N \sum_{t=0}^T \underbrace{\left( \frac{v_{n-1}(t) - v_n(t)}{v_{\max}} \right)^2}_{\text{safety}} \cdot P_{\text{safety}} + \underbrace{\left( \frac{\Delta x_n(t) - \frac{1}{M} \sum_{m=1}^M \Delta x_m(t)}{\Delta x_{\max}} \right)^2}_{\substack{\text{efficiency} \\ \text{Cte non-autonomous}}} \cdot P_{\text{efficiency}} + \underbrace{\left( \frac{\text{acc}_n(t)}{\text{acc}_{\max}} \right)^2}_{\text{comfort}} \cdot P_{\text{comfort}} \quad (1)$$

Concretely, we specify the cost function formulated in Equation 1 to be minimised, and which is inspired by Wang,<sup>35</sup> since we also consider a safety, efficiency, and comfort cost. However, Wang's cost function<sup>35</sup> is specific to the adaptive cruise control (ACC) algorithm for optimising the acceleration parameters, while our cost function (1) can be seen as more generic and can be used by any car-following model to optimise speed, acceleration, or time headway.

The controller distinguishes "free driving mode" for the leading vehicles of the group of vehicles and "following mode" for all the other conventional and autonomous vehicles. The leader vehicle travels at the user-defined speed input  $v_1$ . In following mode, vehicles try to maintain a desired

<sup>†</sup><http://www.omg.org/spec/OCL/>

**TABLE 1** Optimal parameters obtained from the offline global minimisation of the cost function defined in (1)

Parameters	Normal conditions	Light rain	Heavy rain
Acceleration: $acc$ ( $m/s^2$ )	1.87	1.87	1.48
Time headway: $Th$ (s)	1.60	2.12	2.84

gap (efficiency cost) based on the variation of positions ( $\Delta x$ ) while at the same time avoiding driving too close to the predecessor (safety cost). We also consider the driving comfort by including a comfort cost associated to the acceleration parameter to yield more comfortable accelerations and decelerations.<sup>35</sup>  $v_{max}$ ,  $\Delta x_{max}$ , and  $acc_{max}$  are normalisation constants:  $v_{max}$  is the maximum speed of the current road (usually selected as the speed limit, here, 33 m/s) and thus is context based;  $\Delta x_{max}$  is the maximum desired space headway defined as  $\Delta x_{max} = v_{max} \cdot T + s_0$ , where  $T = 1.6$  seconds is the desired time gap and  $s_0 = 2$  m is the gap in standstill conditions. Finally,  $acc_{max}$  is the maximum acceleration<sup>11,35</sup> that is chosen as 3 m/s<sup>2</sup>.

We also assign weights to each term ( $P_{safety}$ ,  $P_{efficiency}$ , and  $P_{comfort}$ ) to adapt safety, efficiency, and comfort, respectively, to different contexts or drivers' preferences.

The parameters values obtained with the offline global minimisation of the cost function (1), assuming full knowledge of the vehicles' speeds and headways under normal, and light and heavy rain conditions, are shown in Table 1. These values are close to the variations we expected, as defined in the previous section (section 2.2) for realistic behaviours.

However, under realistic conditions, autonomous vehicles will only have a very local knowledge of other vehicles' variables,<sup>36</sup> and the parameter values obtained in this section still need to be estimated as precisely as possible at runtime. To do that, in the next section, we investigate a distributed model predictive controller agent for autonomous vehicles to estimate their behavioural parameters at runtime, based on their available knowledge of the system. We show that autonomous vehicles with the proposed reconfigurable controller agent lead to behaviour similar to that obtained theoretically offline, and thus to that achieved by human drivers, depending on the context.

## 5 | RECONFIGURABLE VEHICLE CONTROLLER AGENT

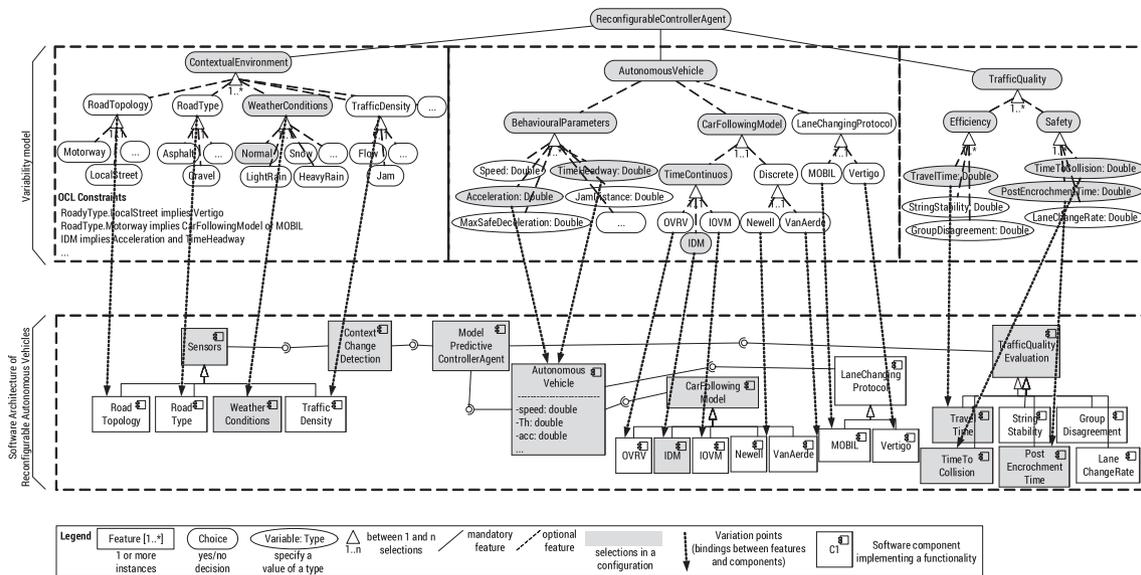
This section describes our reconfigurable controller agent for the autonomous vehicles. First, we present the software architecture of the reconfigurable autonomous vehicle and model the knowledge needed to reconfigure its behaviour. Then, we present a model predictive controller agent to dynamically adapt the behavioural parameters of the vehicle.

### 5.1 | Variability modelling of the autonomous vehicle: traffic context, vehicle behaviour, and traffic quality

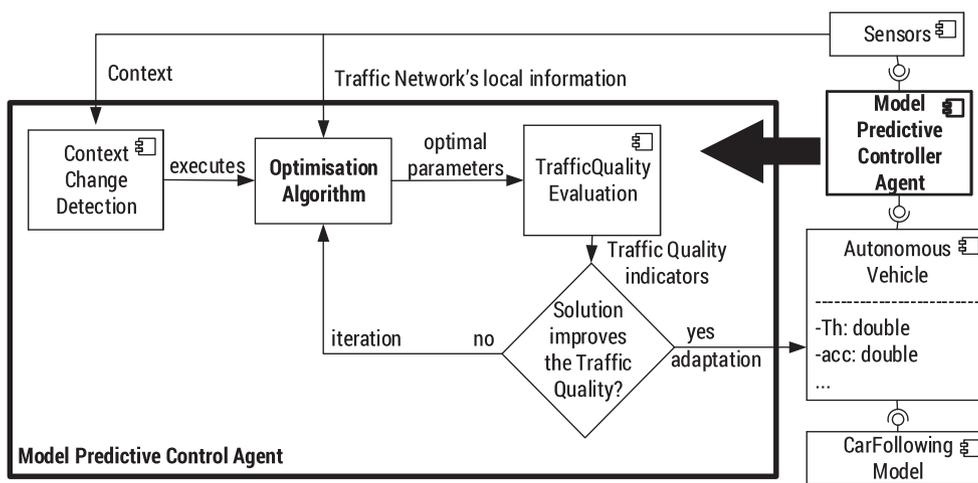
Figure 1 shows the variability model in CVL and the software architecture of the reconfigurable autonomous vehicle connected with the variability model. The variability model (top of Figure 1) specifies (1) the different contexts that can be detected by the sensors of the autonomous vehicle; (2) all the reconfigurable parameters of the autonomous vehicles that affect their behaviour, which includes the behavioural parameters (eg, speed, acceleration, and time headway) and the available car-following models and lane-changing protocols; and (3) the traffic quality indicators to take into account. Each configurable feature is decomposed into choices in the tree, indicating which parameters are optional and which are mandatory. For instance, the `BehaviouralParameters` feature contains the configurable `Speed`, `Acceleration`, `TimeHeadway`, `MaxSafeDeceleration`, and `JamDistance` parameters; although in the example of this, paper we only consider `Acceleration` and `TimeHeadway`.

The variability model also defines groups of features (as the `CarFollowingModel` feature) to specify the car-following model alternatives to be used. Dependencies between features are specified as cross-tree constraints in the OCL. We use OCL constraints to model the relationships between a specific context and the car-following model and lane-changing protocol of the autonomous vehicle to be used (see OCL constraints in Figure 1). For instance, the constraint `RoadType.LocalStreet implies Vertigo` means that when the autonomous vehicle drives through a local street, the lane-changing protocol will be reconfigured to use the `Vertigo` model, which is the appropriate model for that road type.<sup>4</sup>

The bottom half of Figure 1 shows the complete software architecture of the reconfigurable autonomous vehicle. It specifies the software components that implement the functionality of each module (sensors, context change detection, car-following model, lane-changing protocol, parameters of the vehicle itself, and traffic quality evaluation) and the relationships between the components. Therefore, in the DSPL, the variability model describes the potential range of variations that can be produced at runtime for the autonomous vehicle controller. The dynamic variation points, represented as dashed lines in Figure 1, link the features of the variability model and the components in the software architecture of the autonomous vehicle. The variation points model the possible adaptations that the software architecture supports. Specifically, each variation point (dashed line) defines a transformation (eg, object existence, object substitution, and parameter assignment) in the software architecture that will be executed by the CVL engine<sup>37</sup> when a particular configuration is selected in the variability model, adapting the components of the autonomous vehicle.<sup>38,39</sup> For example, when the context changes because the vehicle joins a motorway, a new configuration is generated: The new context is selected in the tree,



**FIGURE 1** Variability modelling of the autonomous vehicles: context, behaviour, and traffic following quality. IDM, Intelligent Driver Model



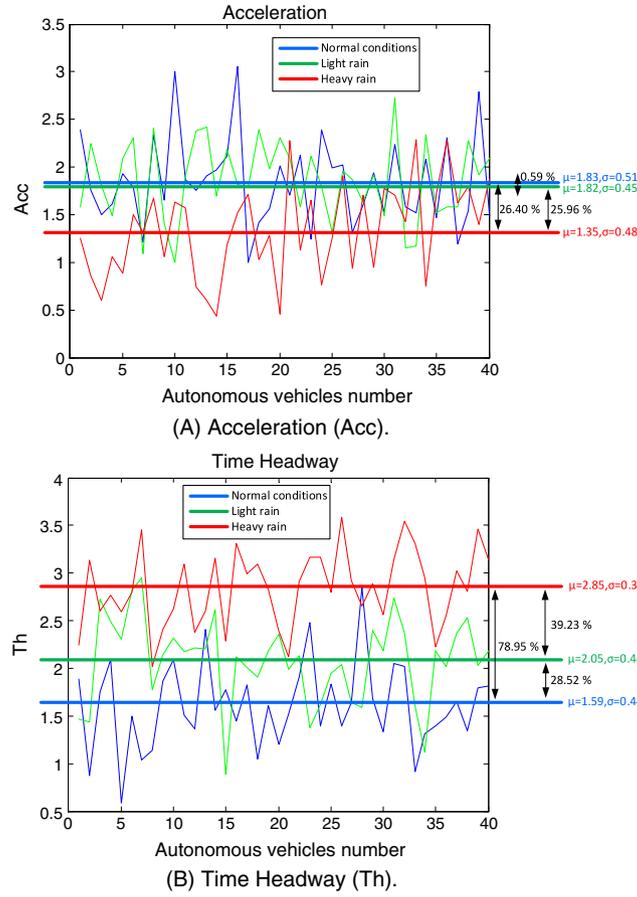
**FIGURE 2** Reconfigurable vehicle controller agent

and the IDM model is automatically selected in the configuration due to the cross-tree constraints defined in OCL. The CVL engine will resolve the variability, changing the IDM component to be the car-following model in the autonomous vehicle. However, although our DSL handles this kind of dynamic reconfiguration of complete modules, this is beyond the scope of this paper, where we mainly focus on the optimisation of the behavioural parameters at runtime.

Each configuration of the controller is a set of choices in the variability model (features in grey in Figure 1). In our approach, we first generate a particular configuration of the reconfigurable vehicle controller agent, which is able to detect changes in weather conditions (rain intensity) and evaluate the traffic quality based on the selected indicators of safety (time to collision, post-encroachment time) and efficiency (travel time). Then, during the simulation, the autonomous vehicle adapts its behaviour at runtime using a model predictive controller agent that optimises its behavioural parameters of the vehicle (ie, acceleration and time headway).

## 5.2 | Model predictive controller agent

Once we have generated an initial configuration of the autonomous vehicle controller, the behavioural parameters of the vehicle can be adapted at runtime to changes in the context. The model predictive controller agent component included in the architecture of the autonomous vehicles is in charge of adapting its behavioural parameters. Figure 2 details this component.



**FIGURE 3** Distributions of the parameters obtained at runtime for the autonomous vehicles

$$F_n = \sum_{t=C+X}^{S_t} \underbrace{\left( \frac{v_n(t)}{v_{\max}} \right)^2}_{\text{safety}} \cdot P_{\text{safety}} + \underbrace{\left( \frac{\Delta x_n(t) - \frac{1}{M} \sum_{m=1}^M \sum_{t_a=C}^{C+X} \Delta x_m(t_a)}{\Delta x_{\max}} \right)^2}_{\text{efficiency}} \cdot P_{\text{efficiency}} + \underbrace{\left( \frac{\text{acc}_n(t)}{\text{acc}_{\max}} \right)^2}_{\text{comfort}} \cdot P_{\text{comfort}} \quad (2)$$

The criteria to initiate a reconfiguration of the behavioural parameters is a contextual change. When a contextual change is detected (Context Change Detection) from the information gathered by the sensors, an optimisation algorithm is executed (Optimisation Algorithm in Figure 1).

Autonomous vehicles have only a very local knowledge of other vehicles' variables (eg, positions and speeds). Consequently, we use an optimisation algorithm inspired from a distributed model predictive controller agent for autonomous vehicles to estimate their behavioural parameters at runtime,<sup>35</sup> based on their available knowledge when the context changes. The objective function is defined in (2), similarly to the offline version defined in (1), but requires predicting the trajectory of the vehicle leader ( $v_1$ ). In this way, the optimisation algorithm starts the prediction  $X$  seconds after the context changes ( $t = C$ ) and simulates the trajectory of the leader with the current parameters during the simulation time ( $S_t$ ). The prediction uses the information of the conventional vehicles for the safety term in (2) during the time interval ( $t_a$ ) defined from the context change ( $t_a = C$ ) until the optimisation algorithm starts ( $t_a = C + X$ ). The optimisation algorithm provides the suboptimal parameters values. Then, the global traffic quality of the vehicle network is evaluated (Traffic Quality Evaluation) using the selected indicators to be considered, and when the new predicted parameters for the autonomous vehicles improve the traffic quality, the values are adapted in the Vehicle Controller. The optimisation algorithm is then running continuously to improve the parameters in each iteration.

Figure 3 represents the distributions of the acceleration and time headway parameters obtained at runtime with the online optimisation algorithm. We can observe how the acceleration values are decreased for each autonomous vehicle as expected when the weather conditions change from normal ( $1.83 \text{ m/s}^2$  on average) to light rain ( $1.82 \text{ m/s}^2$ ) and to heavy rain ( $1.35 \text{ m/s}^2$ ). The change in the acceleration parameter from normal conditions to light rain is almost imperceptible (a decrease by 0.59%), while the change from normal conditions or light rain to heavy rain is

**TABLE 2** Ideal parameters vs optimised parameters of the model predictive controller agent

Weather conditions	Acceleration: $acc$ (m/s <sup>2</sup> )		Time headway: $Th$ (s)	
	Ideal	Optimisation	Ideal	Optimisation
Normal conditions	1.87	1.83	1.60	1.59
Light rain	1.87	1.82	2.12	2.05
Heavy rain	1.48	1.35	2.84	2.85

more relevant (a decrease by 26.40% from normal to heavy rain and a decrease of 25.96% from light rain to heavy rain conditions). Regarding the time headway parameter, the values notably increase as the weather conditions change. From normal conditions to light rain, the time headway is increased by 28.52% (from 1.59 to 2.05 s), while the change from normal to heavy rain conditions is substantial, with an increment in 78.95% (from 1.59 to 2.85 s in heavy rain conditions). The change from light rain to heavy rain supposes an increase by 39.23%. In all cases, the average values of the distributions converge to those obtained offline in Section 4, and as shown in the comparison in Table 2.

## 6 | EVALUATION

To evaluate our reconfiguration approach, we compare our reconfigurable vehicle controller agent against autonomous vehicles without reconfiguration and evaluate the traffic quality of the vehicle network. Section 6.1 describes the 3 traffic quality indicators we use in our evaluation. The experimentation set-up is described in section 6.2 and the results in section 6.3.

### 6.1 | Traffic quality indicators

We use the following traffic quality indicators to evaluate safety and efficiency of the vehicle network.

#### 6.1.1 | Indicators of traffic safety

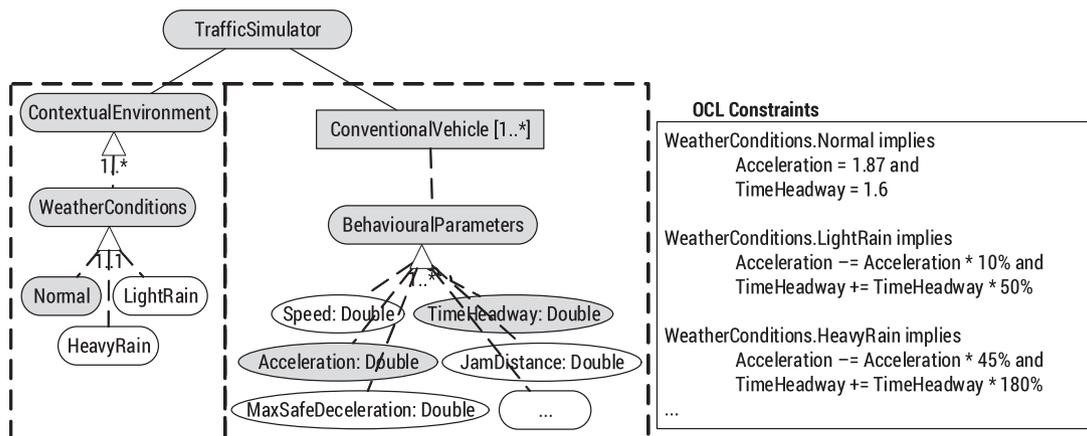
- **Time to collision (TTC):** This is the estimated time at which a collision will occur if 2 vehicles keep the same trajectories and speeds<sup>40</sup>:

$$TTC = \frac{\Delta x_i - l_i}{\Delta v_i} = \frac{x_i - x_{i-1} - l_i}{v_i - v_{i-1}}, \quad (3)$$

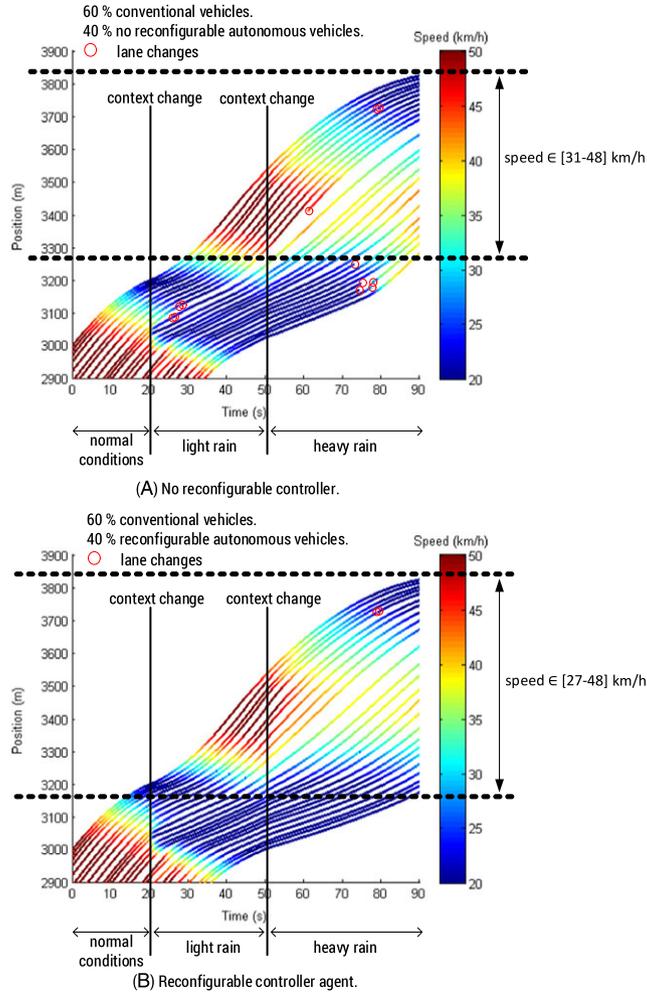
where  $\Delta x_i$  and  $\Delta v_i$  are the difference of position and speed, respectively, between the vehicle  $i$  and its predecessor, and  $l_i$  is the length of the vehicle  $i$ . A time to collision of 0 second means a direct collision.

- **Post-encroachment time (PET):** This is the time difference between 2 vehicles passing a common spatial zone. It represents a direct collision risk,<sup>40,41</sup> and a PET of 0 s is equivalent to a collision, so the objective is to maximise the post-encroachment time:

$$PET = \frac{\Delta x_i - l_i}{v_i} = \frac{x_i - x_{i-1} - l_i}{v_i}. \quad (4)$$



**FIGURE 4** Simulating conventional vehicle behaviour



**FIGURE 5** Vehicles trajectories obtained without/with the proposed reconfigurable controller agent

### 6.1.2 | Indicators of traffic efficiency

- **Travel time (TT):** This is the total time spent by all vehicles in the network. A lower travel time improves the efficiency of the network. It is calculated by summing all the individual times spent  $t_i$  according to the total number of vehicles  $N$ :

$$TT = \sum_{i=1}^N t_i. \quad (5)$$

## 6.2 | Experimentation set-up

We consider a mixed-traffic scenario with conventional and autonomous vehicles, with changes to rainy conditions. The experiments simulate a 1-km 4-lane highway (1 way) scenario for 90 seconds. We simulate 100 vehicles that are spawned at the start of either lane at a configurable rate with time between vehicles following a Poisson distribution. Forty percent of the vehicles are autonomous vehicles with our reconfigurable controller agent assigned, while the other 60% are conventional vehicles.

We also need to simulate the conventional vehicles, representing human drivers. We simulate them using a variability model to represent their configurations and using OCL constraints to give predefined values to their parameters. (Figure 4). Under normal conditions, we fit the parameter  $acc$  (acceleration) with a normal distribution with mean  $\mu = 1.87 \text{ m/s}^2$  and standard deviation of  $\sigma = 0.45$ . The parameter  $Th$  (time headway) is fit with a lognormal distribution with  $\mu = 1.6s$ , and  $\sigma = 0.65$ .<sup>15</sup> In light and heavy rain conditions, we set the parameters of the conventional vehicles as defined in section 2.2.

The same initial positions of the autonomous vehicles and the same leaders' speed inputs are used for this particular experiment. In this paper, we focus on this specific experiment to demonstrate the significance of the reconfiguration approach; however, other simulations with different proportion of autonomous vehicles, and initial parameters were investigated, observing the same tendency in the simulations.<sup>‡</sup>

<sup>‡</sup>The experimental data and the files used to perform the simulations are available in: <http://caosd.lcc.uma.es/ModelPredictiveControlAgent.zip>.

Once a vehicle is created, the controller (using the IDM car-following model) is started that drives the vehicle to the other side of the highway, changing lanes if necessary. At the beginning of the simulation, under normal weather conditions, the leader vehicles travel at speeds varying around  $v_{max}/2$ , while the follower vehicles try to maintain the desired speed, acceleration, and time headway. Then, at 20 seconds of simulation, we introduce a contextual change simulating light rain. The context change is simulated by selecting the desired weather condition in the variability model. Conventional vehicles' behaviour is adapted in accordance with the OCL constraints defined in Figure 4, while the autonomous vehicle behaviours are adapted using our reconfiguration approach as explained in Section 5. Later, at 50 seconds, we introduce another contextual change simulating heavy rain intensity, and vehicles' behavioural parameters are adapted again. We also introduce some manual perturbations (eg, acceleration and deceleration variations) to the leading vehicle trajectory to excite the system and to provoke some variations in the acceleration patterns of the following vehicles. Simulations of autonomous vehicles without reconfiguration means that the used controller does not adapt the vehicle's behaviour to contextual changes.

### 6.3 | Results

In this section, we discuss the results of our evaluation that demonstrate the benefits of our reconfigurable model predictive controller agent.

The trajectories of the vehicles are shown in Figure 5, (A) for mixed traffic with autonomous vehicles without reconfiguration and (B) using the reconfigurable controller agent. These figures represent the speeds (in colour) and positions (y-axis) of the vehicles at each time step (x-axis) for one of the highway lanes. The most left trajectory represents the trajectory of the leading vehicle, while the rest of lines represent the trajectories of both conventional and autonomous vehicles that "follow" the behaviour of the leading vehicle.

We observe that, when the context changes, reconfigurable vehicles adapt their behaviour to slow the speed down and to increase the distance with the predecessor (time headway) due to the worsening weather conditions. For instance, in heavy rain conditions, the speed of the reconfigurable vehicles is lower (between 27 and 48 km/h) than autonomous vehicles without reconfiguration (between 31 and 48 km/h). This decrease of speed implies smoother trajectories (trajectories further spaced) when (B) using our reconfigurable controller agent, contrary to (A) autonomous vehicles without reconfiguration. Note the white spaces between trajectories for the first 10 vehicles in heavy rain conditions in Figure 5B in comparison with (A). Moreover, we can observe that the number of lane changes is reduced (incomplete trajectories and gaps in the same trajectory highlighted

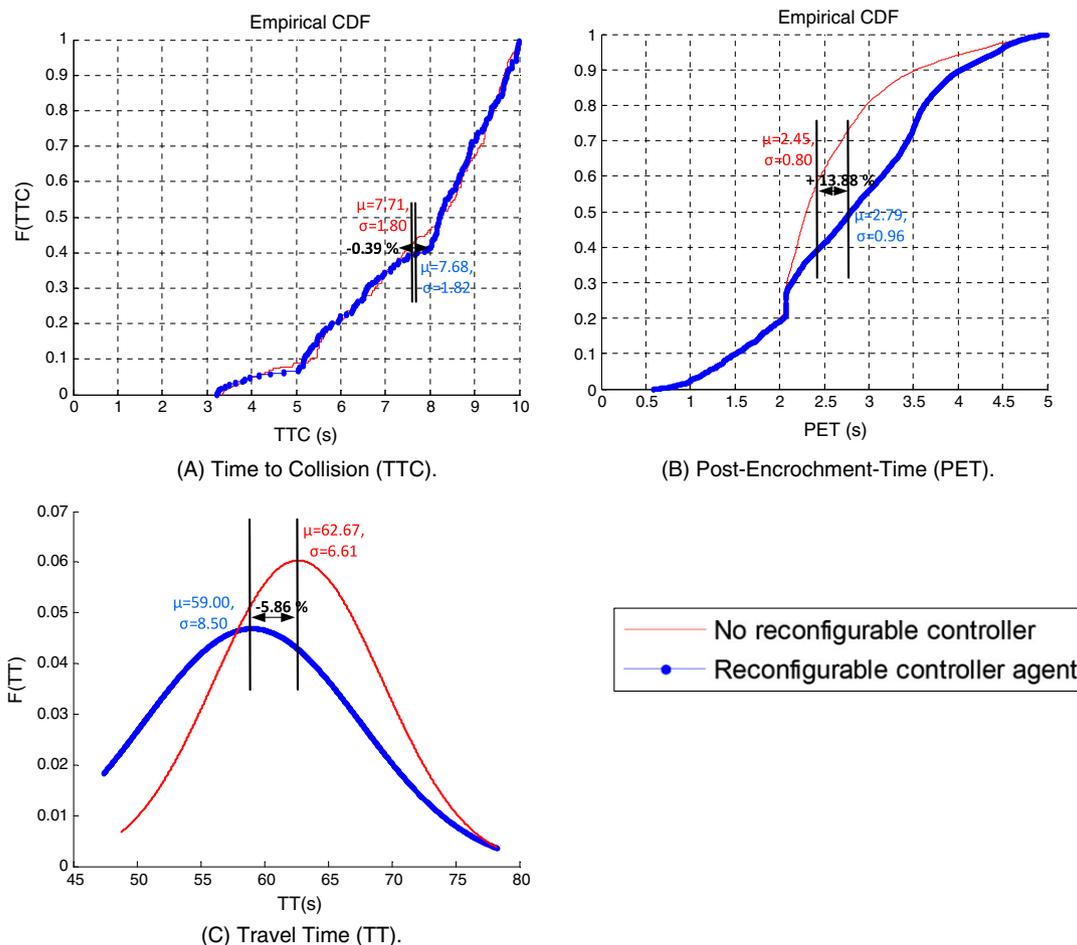


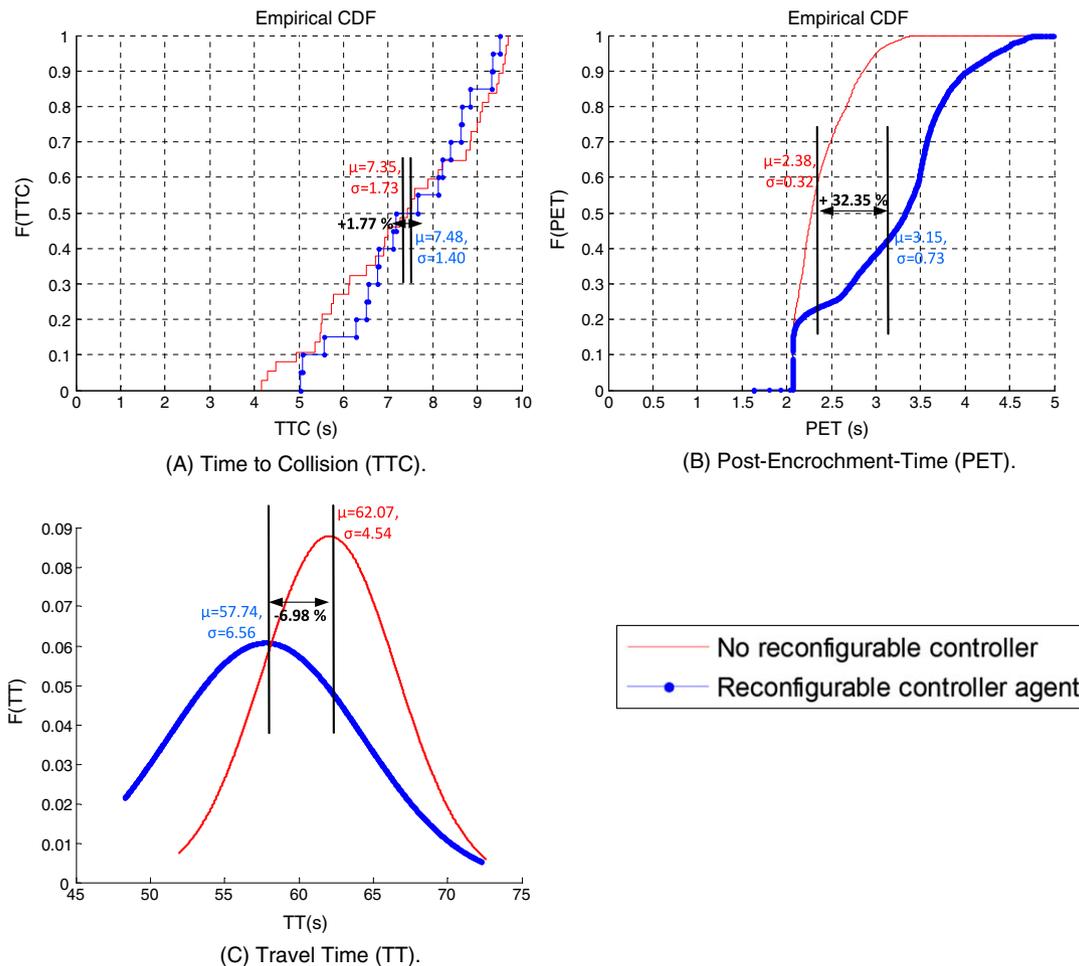
FIGURE 6 Indicators of traffic quality for the entire fleet of vehicles

in a red circle), from 12 lane changes in the traffic network with autonomous vehicles without reconfiguration to only 2 lane changes in the traffic network with reconfigurable autonomous vehicles. Reducing the number of lane changes means that vehicles are more dispersed over the network, maintaining more distance between vehicles and avoiding perilous manoeuvres such as overtaking.

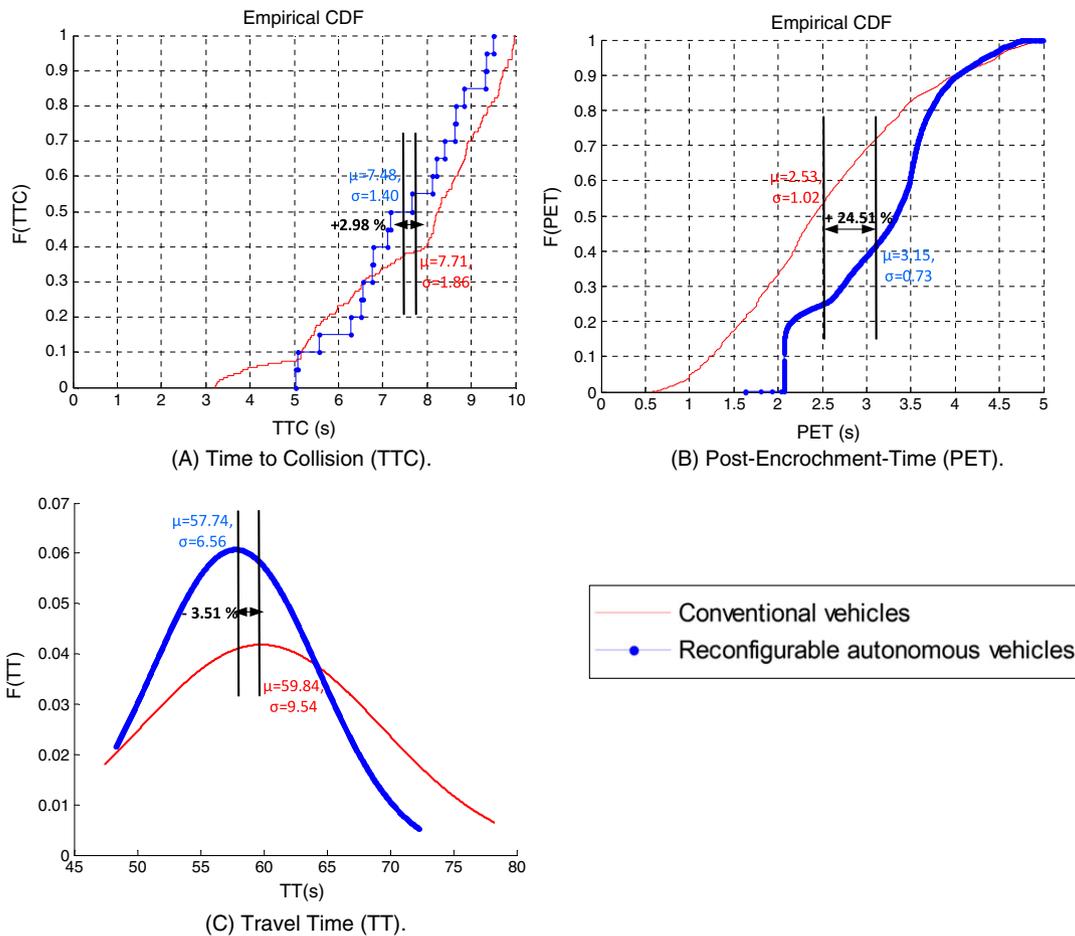
Figure 6 shows the traffic quality of the complete vehicle network. We compare the statistic distributions of the traffic indicators values for (A) TTC, (B) PET, and (C) travel time, for when the mixed-traffic network (ie, conventional and autonomous vehicles sharing the network) includes autonomous vehicles using the reconfigurable controller agent (blue line) and when the mixed-traffic network includes autonomous vehicles without reconfiguration (red line). In the case of safety, the TTC indicator does not present too much variation: 0.39%, from 7.71 seconds on average in the network without reconfiguration to 7.68 seconds in the network with the reconfigurable autonomous vehicles, although the PET increases by 13.88% on average (from 2.45 to 2.79 s), improving the safety of the network. Furthermore, the global efficiency of the vehicle network is improved by reducing the travel time by 5.86% (from 62.67 to 59 s on average), despite the fact that vehicles are more dispersed over the network.

Figure 7 considers the traffic quality indicators calculated only for the autonomous vehicles instead of for the global network. This allows us to compare only the autonomous vehicles with and without reconfiguration. The benefits of using the reconfigurable controller agent are more visible: Both safety and efficiency are improved for the autonomous vehicles. The (A) TTC indicator is increased from 7.35 to 7.48 seconds (1.77%), while (B) PET is increased from 2.38 to 3.15 seconds (32.35%). The (C) TT indicator is also improved by decreasing the travel time from 62.07 to 57.74 seconds (6.98%).

Finally, we also consider the traffic quality indicators only for the conventional vehicles in comparison with the reconfigurable autonomous vehicles (Figure 8). In this case, for safety, conventional vehicles present a TTC of 7.71 seconds, and a PET of 2.53 seconds on average, versus a TTC of 7.48 seconds, and a PET of 3.15 seconds for the reconfigurable autonomous vehicles. A better traffic safety is observed: There is a 2.98% increase of the TTC and a 24.51% increase of the PET. For efficiency, the TT indicator also shows that autonomous vehicles behave in more similar way as a whole than conventional vehicles—ie, the standard deviation of the TT indicator is less for autonomous vehicles ( $\sigma_{TT} = 6.56$ ) than for the conventional vehicles ( $\sigma_{TT} = 9.54$ ). Moreover, autonomous vehicles are more efficient than conventional, decreasing travel time by 3.51%—ie, from 59.84 seconds on average for conventional vehicles vs 57.74 seconds of the reconfigurable autonomous vehicles.



**FIGURE 7** Indicators of traffic quality for the autonomous vehicles with and without reconfiguration



**FIGURE 8** Comparison of traffic quality indicators between the reconfigurable autonomous vehicles and the conventional vehicles

## 7 | CONCLUSIONS

Autonomous vehicles with our reconfigurable controller agent present a similar behaviour to that achieved by human drivers when the weather conditions change. As the intensity of the rain develops, acceleration is reduced up to 25.96% and time headway is increased up to 78.95% under heavy rain, which are very close to the expected variations for human drivers with decrements in acceleration up to 20.86% and increments in time headway up to 77.50%.

Moreover, we have demonstrated that autonomous vehicles with the reconfigurable controller agent increase traffic-flow efficiency and safety under changing adverse weather conditions. In general, reconfigurable autonomous vehicles present a safer driving behaviour than autonomous vehicles without reconfiguration, which is critical when the weather is changing from normal to rainy conditions. We observe a reduction in the rate of lane changes, and an increase in the time to collision and post-encroachment time indicators. The efficiency of the vehicle network, measured by the travel time, is also improved, which is critical when the weather is changing from rainy to normal conditions.

Our reconfiguration approach is easily extensible to dynamically change the software architecture of the autonomous vehicles thanks to the use of DSPLs. In this way, with our approach, it is possible not only to tune car-following model parameters or any other models of the vehicle dynamics but also to dynamically change the model used by the vehicle controller at runtime, for instance, to adapt it to contextual changes relating to the infrastructure. The DSPL approach also allows us to include the specification of the cost function as part of the SPL to customize it for each different model and for different selections of traffic quality indicators.

## ACKNOWLEDGEMENTS

This work is supported by the projects MagicP12-TIC1814 and HADASTIN2015-64841-R (cofinanced by FEDER funds).

## ORCID

José Miguel Horcas  <http://orcid.org/0000-0002-7771-0575>

## REFERENCES

1. Badeig F, Adam E, Mandiau R, Garbay C. Analyzing multi-agent approaches for the design of advanced interactive and collaborative systems. *J Ambient Int Smart Environ*. 2016;8(3):325-346.
2. Corcoba-Magaña V, Muñoz-Organero M, Pañeda XG. Prediction of motorcyclist stress using a heart rate strap, the vehicle telemetry and road information. *J Ambient Int Smart Environ*. 2017;9(5):579-593.
3. Kesting A, Treiber M, Helbing D. General lane-changing model mobil for car-following models. *Transp Research Record: J Transp Res Board*. 2007;1999:86-94.
4. Martin CFS. Group Communication for cooperative automated vehicles. *PhD thesis*: Trinity College Dublin; 2013.
5. Hoogendoorn RG, Tamminga G, Hoogendoorn SP, Daamen W. Longitudinal driving behavior under adverse weather conditions: adaptation effects, model performance and freeway capacity in case of fog. In: 13th International IEEE Conference on Intelligent Transportation Systems (ITSC); 2010; Funchal, Madeira Island, Portugal. 450-455.
6. Soria I, Elefteriadou L, Kondyli A. Assessment of car-following models by driver type and under different traffic, weather conditions using data from an instrumented vehicle. *Simul Model Pract Theory*. 2014;40:208-220.
7. Hallsteinsen S, Hinchey M, Park S, Schmid K. Dynamic software product lines. *Computer*. 2008;41(4):93-95.
8. Metzger A, Pohl K. Software product line engineering and variability management: achievements and challenges. In: Proceedings of the on Future of Software Engineering; 2014; Hyderabad, India. 70-84.
9. Pohl K, Böckle G, van der Linden FJ. *Software Product Line Engineering: Foundations, Principles and Techniques*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.; 2005.
10. Antsaklis PJ, Passino KM, Wang SJ. An introduction to autonomous control systems. *IEEE Control Syst*. 1991;11(4):5-13.
11. Kesting A, Treiber M, Helbing D. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity; 2009.
12. Gipps PG. A behavioural car-following model for computer simulation. *Transp Res Part B: Method*. 1981;15(2):105-111.
13. Yu S, Liu Q, Li X. Full velocity difference and acceleration model for a car-following theory. *Commun Nonlinear Sci Numer Simul*. 2013;18(5):1229-1234.
14. Monteil J, O'Hara N, Cahill V, Bouroche M. Real-time estimation of drivers' behaviour. In: IEEE 18th International Conference on Intelligent Transportation Systems (ITSC); 2015; Las Palmas de Gran Canaria, Canary Islands, Spain. 2046-2052.
15. Punzo V, Montanino M, Ciuffo B. Do we really need to calibrate all the parameters? variance-based sensitivity analysis to simplify microscopic traffic flow models. *IEEE Trans Int Transp Syst*. 2015;16(1):184-193.
16. Ghezzi C. Of software and change. *J Software: Evol Process*. 2017;29(9):e1888-n/a. e1888 smr.1888.
17. Behere S, TÄürngren M. A functional reference architecture for autonomous driving. *Inf Software Technol*. 2016;73(Supplement C):136-150.
18. Haugen Ø, Møller-Pedersen B, Oldevik J, Olsen GK, Svendsen A. Adding standardized variability to domain specific languages. In: International Software Product Line Conference, SPLC; 2008; Limerick, Ireland. 139-148.
19. Billot R, El Faouzi NE, De Vuyst F. Multilevel assessment of the impact of rain on drivers' behavior Standardized methodology and empirical analysis. *Transp Res Record: J Transp Res Board*. 2009;2107:134-142.
20. Edwards JB. Speed adjustment of motorway commuter traffic to inclement weather. *Transp Res Part F: Traffic Psychology Behav*. 1999;2(1):1-14.
21. Gallen R, Hautiere N, Glaser S. Advisory speed for intelligent speed adaptation in adverse conditions. In: The Intelligent Vehicles Symposium (IV'10); 2010; La Jolla, CA, USA. 107-114.
22. Hogema JH. Effects of rain on daily traffic volume and on driving behaviour. Technical report; 1996.
23. Rakha H, Farzaneh M, Arafteh M, Sterzin E. Inclement weather impacts on freeway traffic stream behavior. *Transp Res Record: J Transp Res Board*. 2008;2071:8-18.
24. Agarwal M, Maze TH, Souleyrette R. Impacts of weather on urban freeway traffic flow characteristics and facility capacity. In: Proceedings of the 2005 Mid-Continent Transportation Research Symposium; 2005; Ames, Iowa, USA. 18-19.
25. Asamer J, van Zuylen HJ, Heilmann B. Calibrating car-following parameters for snowy road conditions in the microscopic traffic simulator vissim. *Int Transport Syst, IET*. 2013;7(1):114-121.
26. El Faouzi NE, Billot R, Nurmi P, Nowotny B. Effects of adverse weather on traffic and safety: state-of-the-art and a European initiative. In: 15th International Road Weather Conference; 2010; Quebec, Canada.
27. Jung S, Qin X, Noyce D. Modeling highway safety and simulation in rainy weather. *Transp Res Record: J Transp Res Board*. 2011;2237:134-143.
28. Zhao Y, Sadek A, Fuglewicz D. Modeling the impact of inclement weather on freeway traffic speed at macroscopic and microscopic levels. *Transp Res Record: J Transp Res Board*. 2272;2012:173-180.
29. Cetina C, Fons J, Pelechano V. Applying software product lines to build autonomic pervasive systems. In: Software Product Line Conference (SPLC); 2008; Limerick, Ireland. 117-126.
30. Cabot J, Gogolla M. *Object Constraint Language (OCL): A Definitive Guide*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012:58-90.
31. Pascual GG, Pinto M, Fuentes L. Self-adaptation of mobile systems driven by the common variability language. *Future Gener Comput Syst*. 2015;47:127-144. Special Section: Advanced Architectures for the Future Generation of Software-Intensive Systems.
32. Hallsteinsen S, Geihs K, Paspallis N, et al.. A development framework and methodology for self-adapting applications in ubiquitous computing environments. *J Syst Software*. 2012;85(12):2840-2859.
33. Gamez N, Fuentes L. Architectural evolution of famiware using cardinality-based feature models. *Inf Software Technol*. 2013;55(3):563-580. Special Issue on Software Reuse and Product Lines.
34. Ali GGMD, Chong PHJ, Samantha SK, Chan E. Nawaz Efficient data dissemination in cooperative multi-RSU vehicular ad hoc networks (VANETs). *J Syst Software*. 2016;117(Supplement C):508-527.
35. Meng W. *Generic Model Predictive Control Framework for Advanced Driver Assistance Systems*: TU Delft, Delft University of Technology; 2014.

36. Sucasas V, Mantas G, Saghezchi FB, Radwan A, Rodriguez J. An autonomous privacy-preserving authentication scheme for intelligent transportation systems. *Comput Secur.* 2016;60(Supplement C):193-205.
37. Horcas JM, Pinto M, Fuentes L. Extending the common variability language (CVL) engine: a practical tool. In: Proceedings of the 21st International Systems and Software Product Line Conference, SPLC; 2017; Sevilla, Spain. 32-37. Best Tool Award.
38. Horcas JM, Pinto M, Fuentes L. Injecting quality attributes into software architectures with the common variability language. In: 17th International Symposium on Component-Based Software Engineering (CBSE); 2014; Lille, France. 35-44.
39. Horcas JM, Pinto M, Fuentes L. An automatic process for weaving functional quality attributes using a software product line approach. *J Syst Software.* 2016;112:78-95.
40. Gettman D, Head L. Surrogate safety measures from traffic simulation models. *Transp Res Record: J Transp Res Board.* 2003;1840:104-115.
41. Minderhoud MM, Bovy PHL. Extended time-to-collision measures for road traffic safety assessment. *Accid Anal Prev.* 2001;33(1):89-97.